BLM_314 Algoritma Analizi Dersi Proje Raporu

Erdoğan TURPCU
19360859076@ogrenci.btu.edu.tr

1. Program İçin Gerekli Kurulumların Yapılması

Program Python dili ile geliştirilmiş olup programın çalışabilmesi için bilgisayarınızda Python 3 ve bir Python arayüz kütüphanesi olan PyQt5 kurulu olmalıdır.

a) Python 3 kurmak için: https://www.python.org/downloads/ sitesini ziyaret edebilirsiniz.

b) PyQt5 Kütüphanesini kurmak için işletim sisteminizin terminalini açarak; `pip3 install pyqt5` komutunu yazmanız yeterli olacaktır.

2. Programın Çalışma Şekli

Program, iki tanesi arama algoritması ve 7 tanesi sıralama algoritması olmak üzere toplam 9 adet algoritma içermektedir. Program rastgele sayılar üreterek algoritmaları çalıştırmaktadır. Uygulamada bulunan algoritmalar:

1. Doğrusal Arama Algoritması (Linear Search)
2. İkili Arama Algoritması (Binary Search)
3. Eklemeli Sıralama Algoritması (Insertion Sort)
4. Birleştirmeli Sıralama Algoritması (Merge Sort)
5. Yığın Sıralama Algoritması (Heap Sort)
6. Çabuk Sıralama Algoritması(Quick Sort)
7. Kova Sıralama Algoritması (Bucket Sort)
8. Sayarak Sıralama Algoritması (Counting Sort)
9. Taban Sıralama Algoritması (Radix Sort)

Programı başlatmak için ana dizin içerisindeki *Uygulama.py* isimli dosyayı çalıştırmanız gerekmektedir.

Uygulama başladığı gibi sizden bir algoritma seçmenizi istemektedir. Seçtiğiniz algoritma bir arama algoritması ise isteğe bağlı olarak sizden bulunacak sayı istenecektir.

Yine isteğe bağlı olarak hem arama hem sıralama algoritmaları için rastgele üretilecek sayı miktarı ve üretilecek sayıların aralığı belirlenebilmektedir.

Şekil 1. *Uygulama Başlangıç Ekranı*

Şekil 2. *Arama Algoritması ve sayı üretme ekranı*

Algoritmalar çalıştığında sonuç ekranında, eğer çalıştırılan algoritma arama algoritması ise:

a) Kullanılan algoritma
b) Aranılan sayı
c) Kullanılan veri seti
d) Aranılan sayının bulunup bulunmadığı, bulundu ise kaçıncı indiste bulunduğu
e) Arama için yapılan işlem sayısı
f) Arama boyunca geçen süre

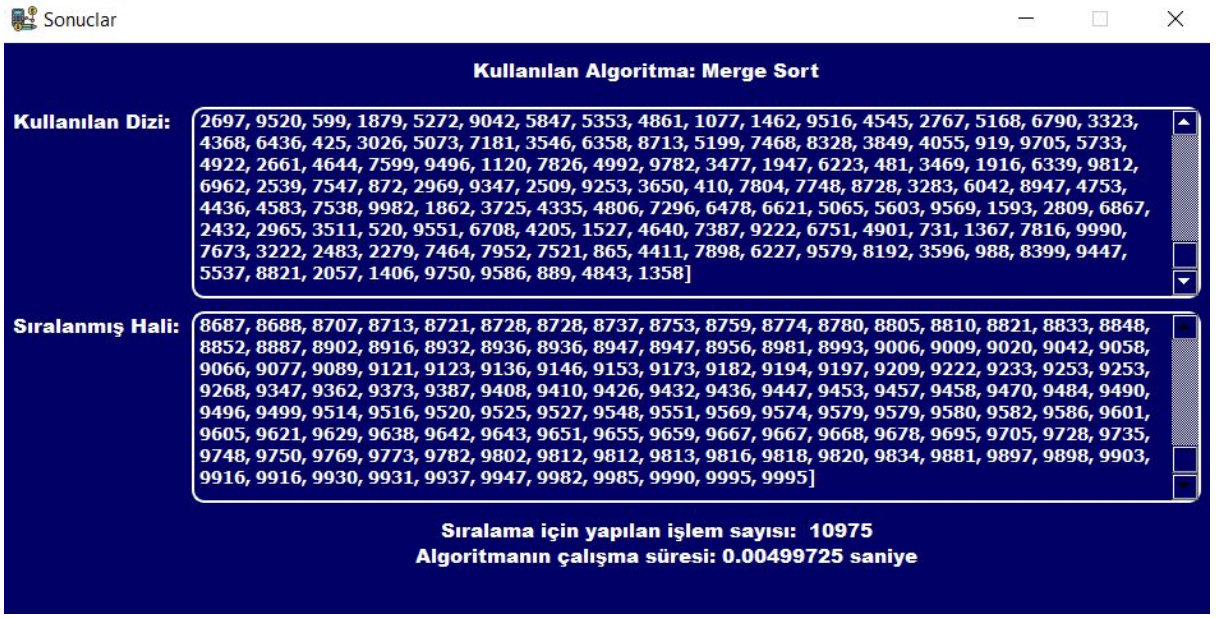
bilgileri yer almakta olup örnek ekran görüntüsü:

Şekil 3. *İkili Arama Algoritması örnek sonuç ekranı*

eğer çalıştırılan algoritma sıralama algoritması ise:

a) Kullanılan algoritma
b) Sıralamadan önceki veri
c) Sıralama sonucu oluşan veri
d) Sıralama için yapılan işlem sayısı
e) Arama boyunca geçen süre

bilgileri yer almakta olup örnek ekran görüntüsü:

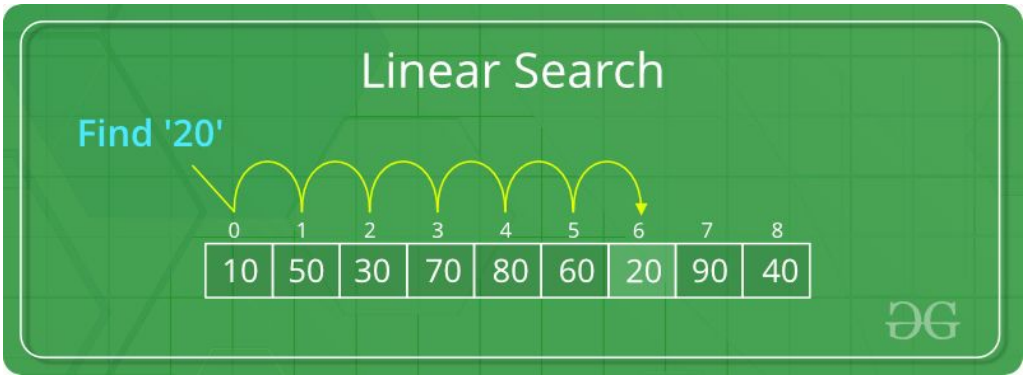


Şekil 4. *Birleştirmeli Algoritması örnek sonuç ekranı*

3. Programda Kullanılan Algoritmalar Ve Ayrıntıları

1. *Lineer Arama Algoritması (Linear Search)*

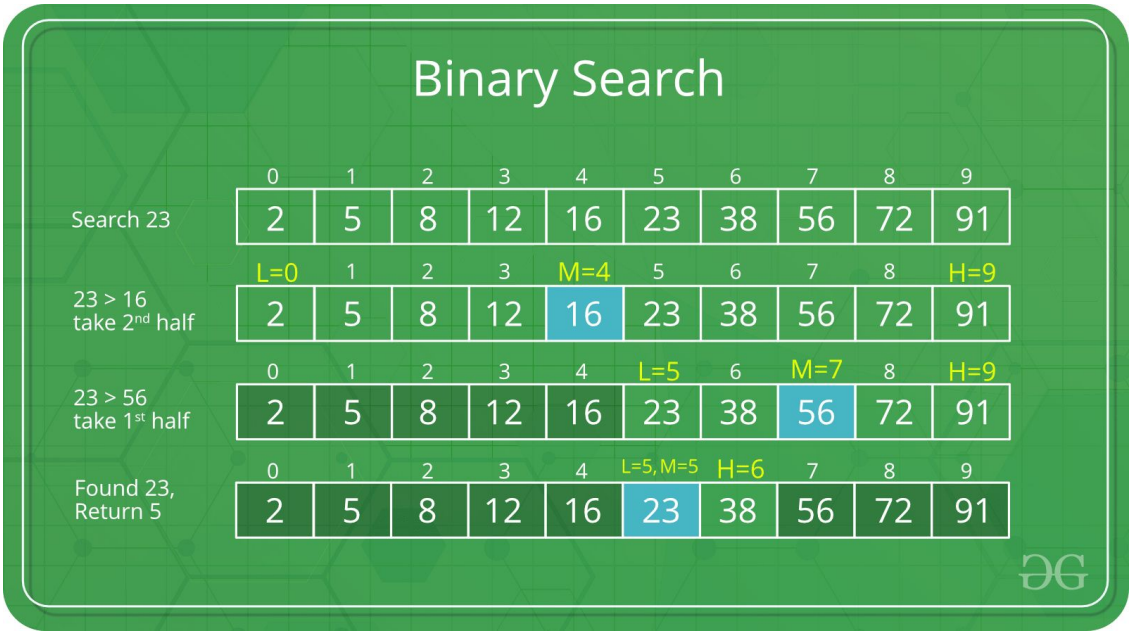
Bir dizi (array) içinde bir verinin olup olmadığını anlamak için kullanılan basit bir algoritmadır. Doğrusal Arama Algoritması aranan veriyi, arrayin ilk öğesinden başlayarak son öğesine doğru, her terimle tek tek karşılaştırır. Bu algoritma, dizi içinde aranan sayıya eşit olan bir terim bulursa onun indisini verir. Aranan veriye eşit olan terim bulamazsa -1 değerini verir. –1 değeri verdiğinde, aranan öğe array içinde değildir.



Şekil 5. *Doğrusal Arama Algoritması*

2. *İkili Arama Algoritması (Binary Search)*

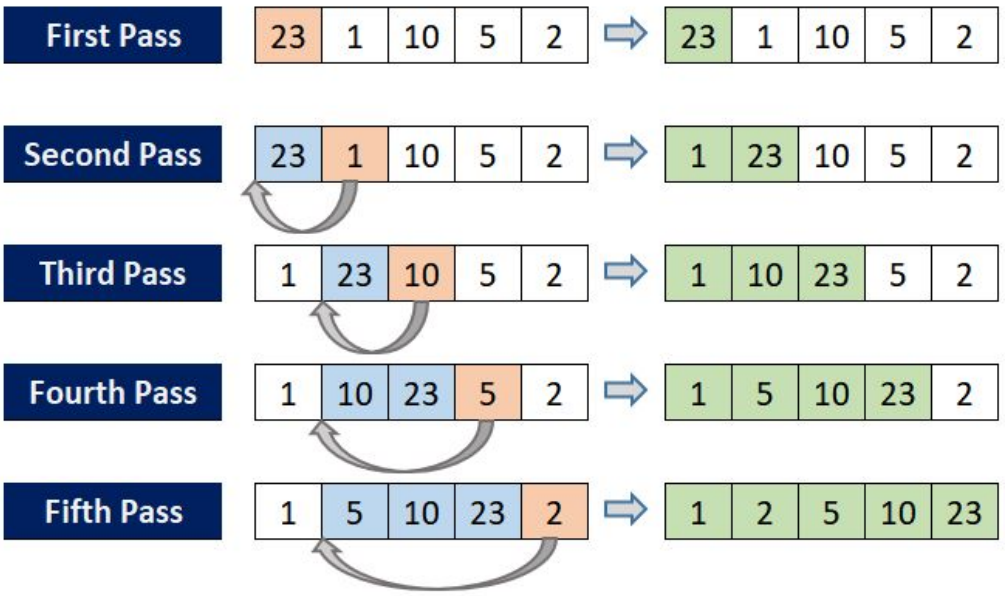
İkili arama algoritması, bir dizide arama yapmaya yarayan bir arama algoritmasıdır. İkili arama algoritmasında, aranan elemanın bulunabilmesi için her seferinde dizinin ortasındaki elemana bakılır. Ortadaki eleman aranan elemana eşit değilse, aranan elemanın bulunduğu diğer yarı alanda arama işlemi tekrar edilir. Bu sayede her adımda arama uzayı yarıya indirilmiş olur.



Şekil 6. *İkili Arama Algoritması*

3. Eklemeli Sıralama Algoritması(Insertion Sort)

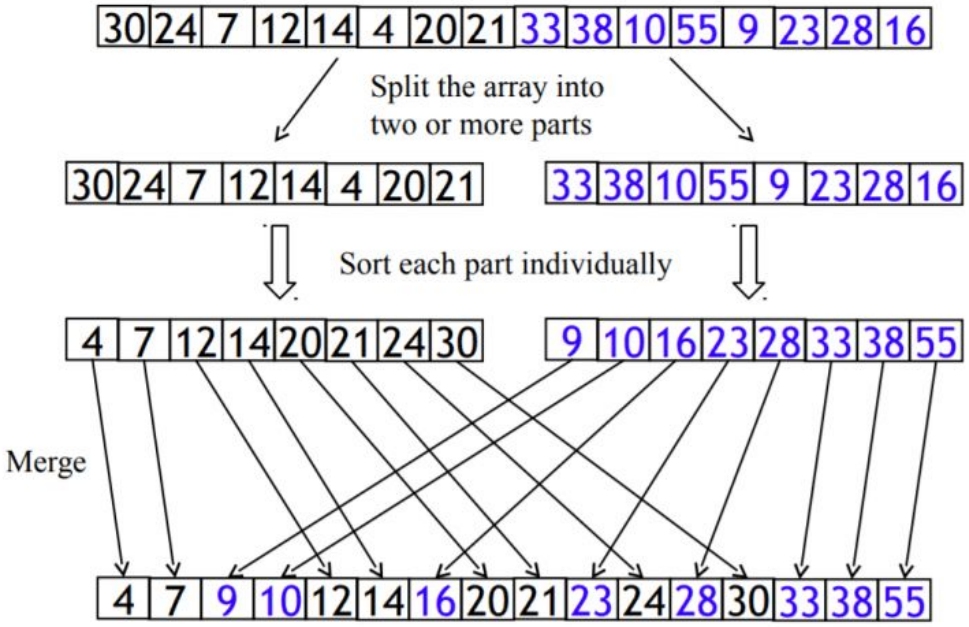
Insertion sort algoritması temel sıralama algoritmalarından bir tanesidir. Algoritmanın mantığına göre elimizdeki A dizisinin elemanları arasında sıralama yapılmak istenildiğinde A[1] indisinden başlanarak önceki elemanlar ile karşılaştırma yapılır. Eğer A[1] indisli eleman kendinden önceki elemanlardan küçük ise yer değiştirme işlemi gerçekleşir ve dizinin başına A[1] elemanı geçer, aksi takdirde yer değiştirme işlemi gerçekleşmez ve bir sonraki elemana bakılır.



Şekil 7. *Eklemeli Sıralama Algoritması*

4. *Birleştirmeli Sıralama Algoritması (Merge Sort)*

Merge sort (Birleştirmeli Sıralama), diziyi ardışık olarak en küçük alt dizilerine kadar yarılayan sonra da onları sıraya koyarak birleştiren özyineli bir algoritmadır. Yarılama işlemi en büyük alt dizi en çok iki öğeli olana kadar sürer. Sonra merge (birleşim) işlemiyle alt diziler ikişer ikişer bölünüş sırasıyla sıralı olarak bir üst dizide birleşir. Süreç sonunda en üstte sıralı diziye ulaşılır.

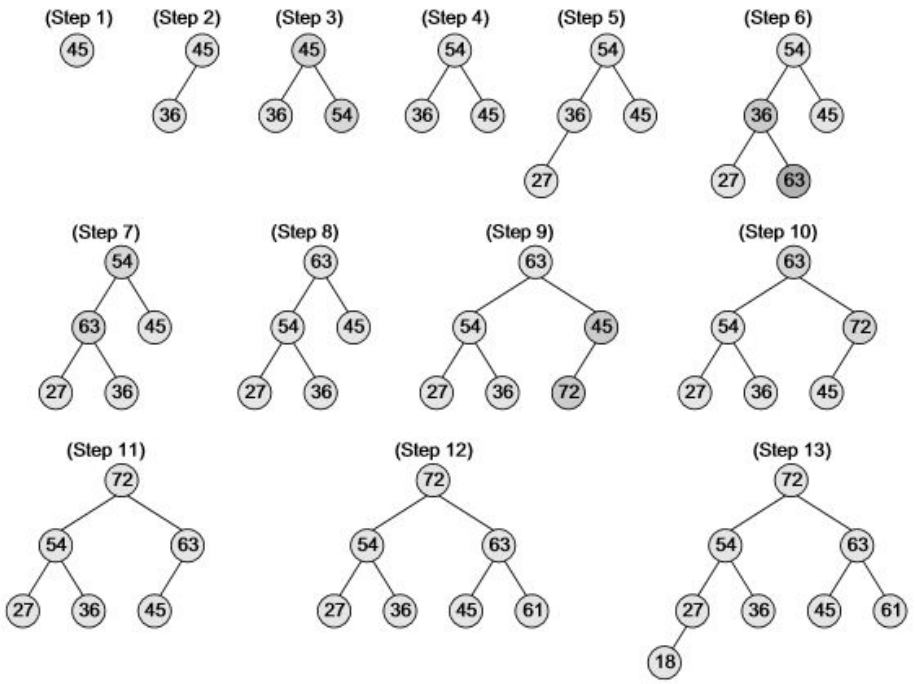


Şekil 8. *Birleştirmeli Sıralama Algoritması*

5. *Yığın Sıralama Algoritması (Heap Sort)*

Heap Sort, her düğümün çocuk düğümlerinin kendinden küçük veya eşit olma kuralını esas alır. Dizinin ilk elemanı her zaman en büyük elemandır. Dizi üzerinde i. eleman ile , çocukları 2*i. ve ((2*i)+1) karşılaştırılıp büyük olan elemanlar yer değiştirilir.

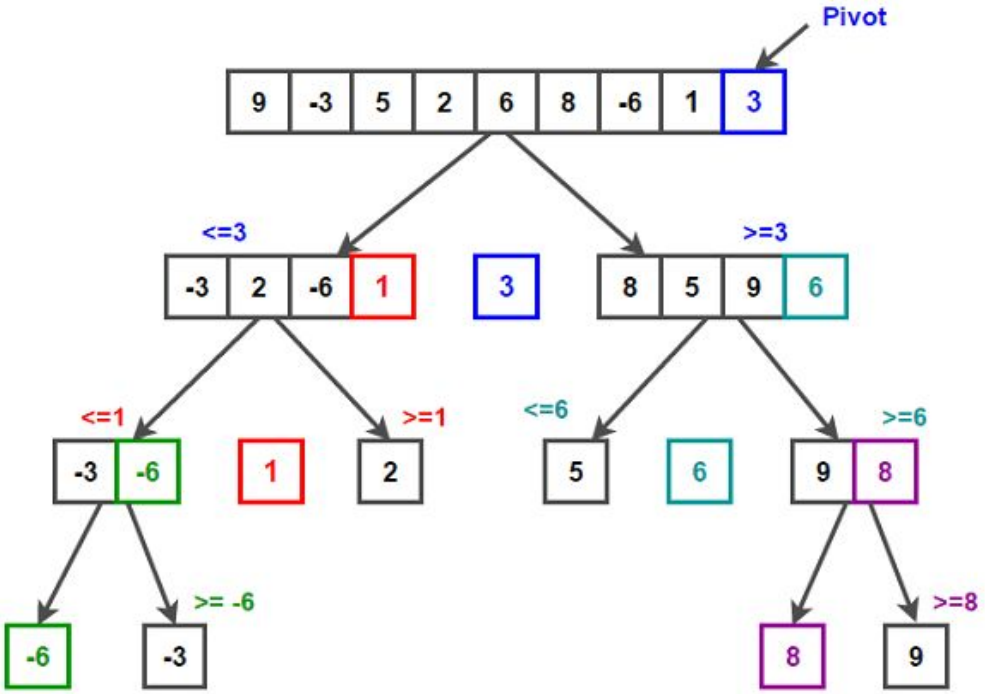
Dizinin son elemanları dizinin ortasındaki elemanların çocuk düğümü olduğundan bu işlem dizinin yarısına kadar yapılır. Elde edilen diziyi sıralamak için ise dizinin ilk elemanı en büyük olduğu bilindiğinde dizini son elemanıyla ilk elemanı yer değiştirilerek büyük eleman sona atılır. Bozulan diziye bu işlemler baştan sona tekrar uygulanır. Dizi boyutu 1 oluncaya kadar işleme devam edilir.



Şekil 9. *Yığın Sıralama Algoritması*

6. *Çabuk Sıralama Algoritması(Quick Sort)*

Bu algoritma, başlarken dizinin terimleri arasından bir terimi pivot olarak seçer. Sonra verilen diziyi üç alt diziye ayrıştırır. Pivot terimden küçük olan terimlerin hepsini (soldaki) birinci alt diziye taşır. İkinci alt dizi sadece pivot terimi tutan tek terimli bir dizidir. Pivot terimden büyük olan terimlerin hepsini (sağdaki) ikinci alt diziye taşır. Sonra sol ve sağ alt dizilere aynı ayrıştırma yöntemini, alt diziler tek terimli birer diziye indirgenene kadar uygular ve sıralama işlemi biter. Algoritma özyineli olarak çalışır.

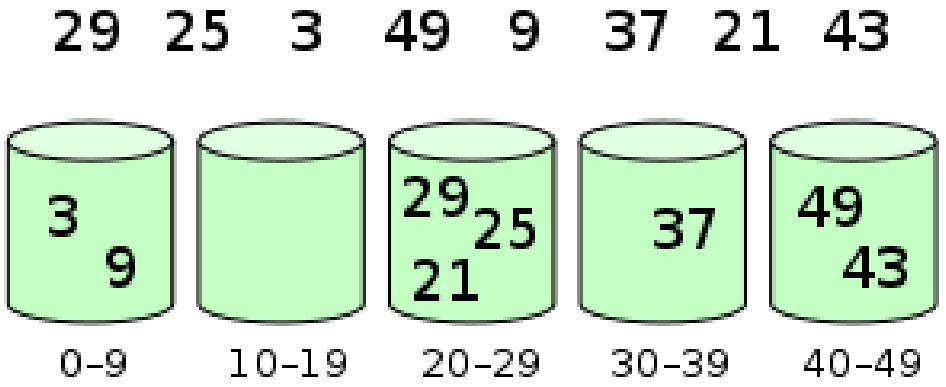


Şekil 10. *Çabuk Sıralama Algoritması*

7. ***Kova Sıralama Algoritması (Bucket Sort)***

Kova Sıralaması (ya da sepet sıralaması), sıralanacak bir diziyi parçalara ayırarak sınırlı sayıdaki kovalara (ya da sepetlere) atan bir sıralama algoritmasıdır. Ayrışma işleminin ardından her kova kendi içinde ya farklı bir algoritma kullanılarak ya da kova sıralamasını özyinelemeli olarak çağırarak sıralanır. Kova sıralaması aşağıdaki biçimde çalışır:

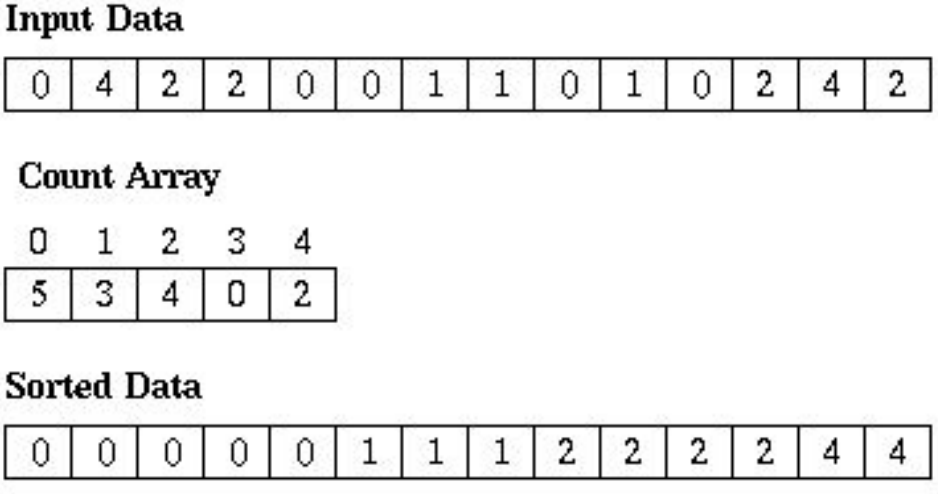
a) Başlangıçta boş olan bir "kovalar" dizisi oluştur.
b) Asıl dizinin üzerinden geçerek her öğeyi ilgili aralığa denk gelen kovaya at.
c) Boş olmayan bütün kovaları sırala.
d) Boş olmayan kovalardaki bütün öğeleri yeniden diziye al.



Şekil 11. *Kova Sıralama Algoritması*

8. ***Sayarak Sıralama Algoritması (Counting Sort)***

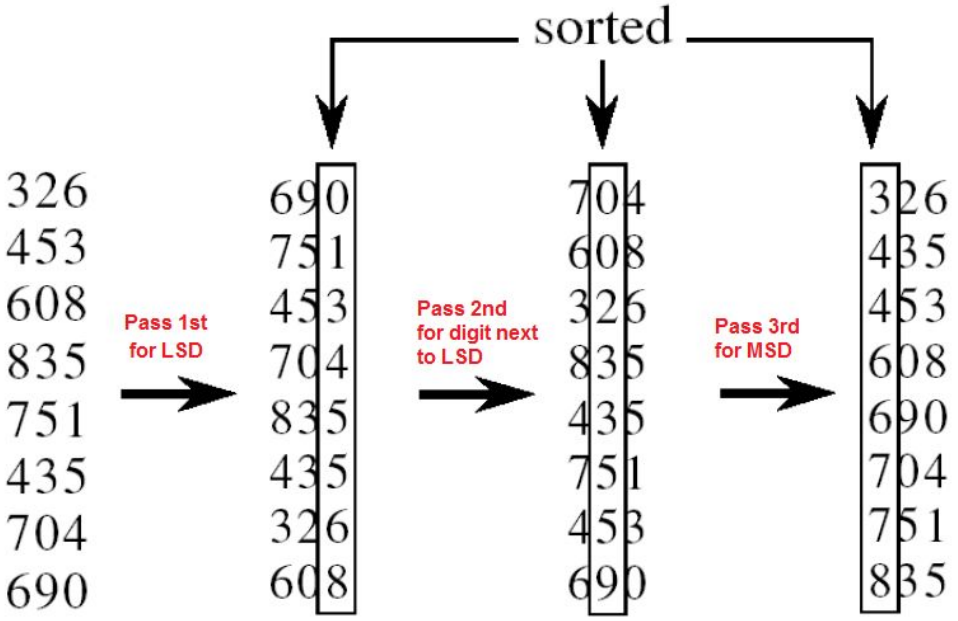
Sayarak sıralama algoritması dizideki değerlerin aralık bilgilerini yeni bir dizi oluşturmak için kullanır. Oluşturulan yeni dizinin her bir satırı ana dizide o satır numarasının değerine sahip ögelerin sayısını gösterir. Yeni dizideki öge değeri sayıları daha sonra ana dizideki tüm değerlerin doğru konuma konulması için kullanılır.



Şekil 12. *Sayarak Sıralama Algoritması*

9. ***Taban Sıralama Algoritması (Radix Sort)***

Sıralanacak verilerin tam sayı olduğu durumlarda kullanılan bu algoritma işlenirken ilk olarak sıralanacak olan veri kümesindeki elemanların en büyük elemanının kaç basamaklı olduğu tespit edildikten sonra sayıların en değersiz olan basamağından itibaren incelenmeye başlanır ve yeni bir diziye yerleştirilir.Bu işlem dizinin en büyük elemanının basamak sayısı kadar tekrar edilir.



Şekil 13. *Taban Sıralama Algoritması*

Programda kullanılan bütün algoritmaların Python dilinde kodlanmış hali ana dizinde bulunan *Algorithms* isimli klasörün içerisinde bulunmaktadır.

4. Kaynaklar

https://www.geeksforgeeks.org/linear-search/
http://mail.baskent.edu.tr/~tkaracay/etudio/ders/prg/dataStructures/searching/LinearSearch.pdf
https://www.geeksforgeeks.org/binary-search/
https://yazilimkaravani.net/ikili-arama-binary-search-algoritmasi/
https://www.alphacodingskills.com/algo/insertion-sort.php
https://yazilimkaravani.net/araya-sokma-insertion-sort-algoritmasi/
https://www.yazilimaktif.com/merge-sort-algoritmasi-birlestirmeli-siralama/
http://www.baskent.edu.tr/~tkaracay/etudio/ders/prg/dataStructures/sorting/MergeSort/MergeSort.pdf
http://furkanalniak.com/siralama-algoritmalari-heap-sort-yigin-siralamasi/
https://deepai.org/machine-learning-glossary-and-terms/quicksort-algorithm
http://mail.baskent.edu.tr/~tkaracay/etudio/ders/prg/dataStructures/sorting/QuickSort/QuickSort.pdf
http://ceng.harran.edu.tr/demoday/2019/ParallelSortingAlgorithms/dokumanlar/RAPOR1.pdf

https://en.wikipedia.org/wiki/Bucket_sort
https://laptrinhx.com/counting-sort-program-in-c-1674313240/
https://www.ybsblog.com/sayarak-siralama-counting-sort-algoritmasi/